Support Vector Machine Classifier with Pinball Loss

Xiaolin Huang, Lei Shi, and Johan A.K. Suykens

***Abstract*—Traditionally, the hinge loss is used to construct support vector machine (SVM) classifiers. The hinge loss is related to the shortest distance between sets and the corresponding classifier is hence sensitive to noise and unstable for re-sampling. In contrast, the pinball loss is related to the quantile distance and the result is less sensitive. The pinball loss has been deeply studied and widely applied in regression but it has not been used for classification. In this paper, we propose a SVM classifier with the pinball loss, called pin-SVM, and investigate its properties, including noise insensitivity, robustness, and misclassification error. Besides, insensitive zone is applied to the pin-SVM for a sparse model. Compared to the SVM with the hinge loss, the proposed pin-SVM has the same computational complexity and enjoys noise insensitivity and re-sampling stability.**

***Index Terms*—classification, support vector machine, pinball loss.**

I. Introduction

SINCE support vector machines (SVM) have been proposed by Vapnik [1] along with other researchers, they have been widely studied and applied in many fields. The basic idea of SVM is trying to maximize the distance between two classes, and the distance between classes is traditionally defined by the closest points. Consider a binary classification problem. We are given a sample set $\mathbf{z} = \{x_i, y_i\}_{i=1}^m$, where $x_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}$. Then $\mathbf{z}$ consists of two classes with the following sets of indices: $\mathbf{I} = \{i \mid y_i = 1\}$ and $\mathbf{II} = \{i \mid y_i = -1\}$. Let $\mathcal{H}$ be a hyperplane given by $w^T x + b = 0$ with $w \in \mathbb{R}^n$, $\|w\| = 1$, and $b \in \mathbb{R}$. We say that $\mathbf{I}$ and $\mathbf{II}$ are separable by $\mathcal{H}$ if for $i = 1, \ldots, m$,

$$\begin{cases} w^T x_i + b > 0, & \forall i \in \mathbf{I}, \\ w^T x_i + b < 0, & \forall i \in \mathbf{II}. \end{cases}$$

In this case, $y_i(w^T x_i + b)$ gives the distance between point x_i and the hyperplane $\mathcal{H}$. Then the distance of each class to hyperplane $\mathcal{H}$ is defined as,

$$\begin{aligned} t_{\mathbf{I}}(w, b) &= \min_{i \in \mathbf{I}} \{y_i(w^T x_i + b)\}, \\ t_{\mathbf{II}}(w, b) &= \min_{i \in \mathbf{II}} \{y_i(w^T x_i + b)\}. \end{aligned}$$

The corresponding classification hyperplane is obtained by

$$\max_{\|w\|=1, b} \quad \{t_{\mathbf{I}}(w, b) + t_{\mathbf{II}}(w, b)\}, \tag{1}$$

which can be equivalently posed into the well-known SVM formulation. From the discussion above, one can see that the result of (1) depends on only a small part of the input data and is sensitive to noise, especially noise around the decision boundary. Consider a one-dimensional example shown in Fig.1. Data of class $+1$ come from distribution $\mathcal{N}(2.5, 1)$, of which the probability density function (p.d.f.) is shown by green line in Fig.1(a). Similarly, $x_i, i \in \mathbf{II} \sim \mathcal{N}(-2.5, 1)$ and the corresponding p.d.f. is shown by red line. The ideal classification boundary is $x = 0$, which can be obtained only when the sampled data satisfy $\min_{i \in \mathbf{I}} x_i = \min_{i \in \mathbf{II}} -x_i$. In other cases, though the sampled data come from the same distribution, the classification result will differ. This observation implies that the result is not stable for re-sampling, which is a common technique for large scale problems. Consider two groups of input data shown in Fig.1(b) and Fig.1(c), where data in two classes are marked by green stars and red crosses, respectively. The positions of $\min_{i \in \mathbf{I}} x_i$, $\max_{i \in \mathbf{II}} x_i$, and the classification boundaries obtained by (1) are illustrated by solid lines. Data in Fig.1(c) can also be regarded as noise corrupted data from Fig.1(b). As illustrated in this example, the classification results of (1) are quite different, showing the sensitivity to noise and instability to re-sampling.

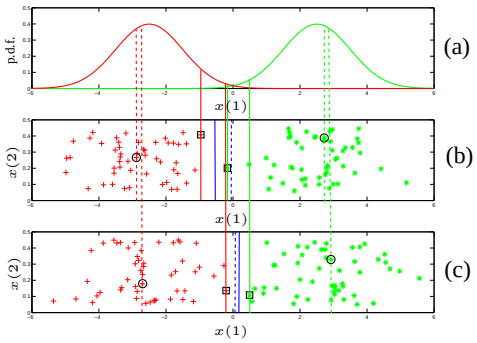


Fig. 1. Data following the p.d.f. shown in (a) are illustrated in (b) and (c), where $x_i, i \in \mathbf{I}$ are marked by green stars and $x_i, i \in \mathbf{II}$ are marked by red crosses. The extreme position in each class and the classification boundaries obtained by (1) are shown by solid lines, while the medium position and the boundaries obtained by (2) with $q = 0.5$ are shown by dashed lines. Though data in (b) and (c) come from the same distribution, the results of (1) are quite different. Data in (c) also can be regarded as noise corrupted data from data in (b), showing the noise sensitivity of (1). Notice that in this problem, only the horizontal position of each point is considered as a feature.

This work was supported by Research Council KUL: GOA/11/05 Ambiorics, GOA/10/09 MaNet, CoE EF/05/006 Optimization in Engineering (OPTEC), IOF-SCORES4CHEM, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects: G0226.06 (cooperative systems and optimization), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine) research communities (WOG: ICCoS, ANMMM, MLDM); G.0377.09 (Mechatronics MPC), G.0377.12 (Structured models), IWT: PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare; Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011); IBBT; EU: ERNSI; ERC AdG A-DATADRIVE-B, FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940); Contract Research: AMINAL; Other: Helmholtz: viCERP, ACCM, Bauknecht, Hoerbiger. L. Shi is also supported by the National Natural Science Foundation of China (11201079). Johan Suykens is a professor at KU Leuven, Belgium.

The authors are all with the Department of Electrical Engineering (ESAT-STADIUS), KU Leuven, B-3001, Leuven, Belgium. L. Shi is also with School of Mathematical Sciences, Fudan University, Shanghai, P.R. China. (e-mails: huangxl06@mails.tsinghua.edu.cn, leishi@fudan.edu.cn, johan.suykens@esat.kuleuven.be).

As mentioned in [2], classification problems may have noise on both y_i and x_i. The noise on y_i is called *label noise* and has been noticed for a long time. The noise on x_i is called *feature noise*, which can be caused by instrument errors and sampling errors. The separating hyperplane obtained by (1) is sensitive to both label noise and feature noise. This is mainly because (1) is trying to maximize the distance between the minimal value of $\{w^T x_i + b\}, i \in \mathbf{I}$ and the maximal value of $\{w^T x_i + b\}, i \in \mathbf{II}$. Some anti-noise techniques have been discussed in [3] [4] [5] and [6]. These methods are based on denoising or weight varying, but the basic idea is still to maximize the distance between the closest points, which is essentially sensitive to noise. Another way of dealing with noise, especially the feature noise, is using robust optimization method to handle the uncertainty, see [2] [7] [8] [9]. One interesting approach was proposed in [10], where the centers of two classes were used to define the distance. Similarly, the means of classes and the total margin were used in [11] and [12], respectively, to construct SVM. In [13] [14] [15], fuzzy and rough sets were introduced into SVM to get less sensitive results. The above methods achieve some success in different applications but generally they lose the elegant formulation of the classical SVM. Meanwhile, additional computation is usually required and the training processes take much more time than the classical SVM.

This paper tries to equip SVM with noise insensitivity and meanwhile preserve the formulation of the classical SVM. For this purpose, we change the idea of (1), i.e., maximizing the shortest distances between two classes, into maximizing the quantile distance. Specifically, we are trying to maximize the sum of the q-lower quantile values of $\{y_i(w^T x_i + b)\}, i \in \mathbf{I}$ and $\{y_i(w^T x_i + b)\}, i \in \mathbf{II}$, respectively. For given w, b, define

$$t^q_{\mathbf{I}}(w,b) = \min^q_{i\in\mathbf{I}}\{y_i(w^T x_i + b)\},$$
$$t^q_{\mathbf{II}}(w,b) = \min^q_{i\in\mathbf{II}}\{y_i(w^T x_i + b)\},$$

where $0 \le q \le 1$ and $\min^q_i(u_i)$ stands for the q-lower quantile of the set $\{u_i\}$. The related classification boundary can be obtained by

$$\max_{\|w\|=1,b} \quad \{t^q_{\mathbf{I}}(w,b) + t^q_{\mathbf{II}}(w,b)\}. \tag{2}$$

From the statistical meaning of quantiles, we expect that (2) is less sensitive to noise and more stable for re-sampling. As an example, the classifiers obtained by (2) with $q = 0.5$ are shown by dashed lines in Fig.1.

Unfortunately, (2) is non-convex and we have to find a convex problem to approach it. For this purpose, the relationship between the hinge loss and (1) is investigated. The hinge loss is defined as,

$$L_{\text{hinge}}(u) = \max\{0, u\}, \forall u \in \mathbb{R}.$$

It is well known that (1) is equal to

$$\min_{w,b}\left\{\frac{1}{2}\|w\|^2, \text{s.t.} \quad \min_i\{y_i(w^T x_i + b)\} = 1\right\}. \tag{3}$$

Then according to the fact that

$$\min_i\{y_i(w^T x_i + b)\} \in \arg\min_{t\in\mathbb{R}} \sum_i L_{\text{hinge}}\left(t - y_i(w^T x_i + b)\right),$$

we can formulate (3) as follows,

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & \textstyle\sum_i L_{\text{hinge}}\left(t - y_i(w^T x_i + b)\right) \\ & \ge \textstyle\sum_i L_{\text{hinge}}\left(1 - y_i(w^T x_i + b)\right), \forall t \in \mathbb{R}. \end{aligned}$$

To deal with the constraint, $\sum_i L_{\text{hinge}}\left(1 - y_i(w^T x_i + b)\right)$ is minimized, which results in the following problem,

$$\min_{w,b} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^m L_{\text{hinge}}\left(1 - y_i(w^T x_i + b)\right). \tag{4}$$

This is actually the well-known SVM with the hinge loss proposed by [1]. In this paper, we call (4) a hinge loss SVM.

Motivated by the link between the hinge loss and the shortest distance, we propose a new SVM classifier with the pinball loss in this paper. The pinball loss is related to quantiles and has been well studied in regression, see [16] for parametric methods and [17] [18] for nonparametric methods. However, the pinball loss has not been used for classification yet. For binary classification, the most widely used loss function is the hinge loss proposed in [1], which results in the hinge loss SVM (4). Besides the hinge loss, the q-norm loss, the Huber loss, and the ℓ_2 loss have also been used in classification, see [1] and [19] for details. For these losses, the bounds of classification error, the learning rates, the robustness and some other properties can be found in [20] [21] and [22]. In this paper, we will use the pinball loss in classification and find that SVM with the pinball loss shares many good properties of the hinge loss SVM. In form, the difference between the hinge loss SVM and the proposed method is that the pinball loss is used instead of the hinge loss. In essence, introducing the pinball loss into classification brings noise insensitivity. The numerical studies will illustrate the performance of using the pinball loss in classification.

The rest of this paper is organized as follows: in Section II, the pinball loss is introduced and a SVM classifier with the pinball loss is proposed. Some properties about the pinball loss are discussed in Section III. Then ε insensitive zone is introduced to the pinball loss for sparsity in Section IV. Section V evaluates the proposed method by numerical experiments. Section VI ends the paper with conclusions.

II. SVM WITH PINBALL LOSS FOR CLASSIFICATION

A. Pinball loss

The pinball loss is given as follows,

$$L_\tau(u) = \begin{cases} u, & u \ge 0, \\ -\tau u, & u < 0, \end{cases}$$

which can be regarded as a generalized ℓ_1 loss. For quantile regression, pinball loss is usually defined as another formulation, see [16] [17], but we can always equivalently set the slope on one side to be 1.

The pinball loss L_τ defines the $\frac{\tau}{1+\tau}$ lower quantile, i.e.,

$$t^{\frac{\tau}{1+\tau}}_{\mathbf{I}}(w,b) = \arg\min_{t\in\mathbb{R}} \sum_{i\in\mathbf{I}} L_\tau\left(t - y_i(w^T x_i + b)\right),$$

and

$$t_{\mathbf{II}}^{\frac{\tau}{1+\tau}}(w,b) = \arg\min_{t\in\mathbb{R}} \sum_{i\in\mathbf{II}} L_\tau\left(t - y_i(w^T x_i + b)\right).$$

Following the method of formulating the hinge loss SVM (4) from problem (1), we set $\tau = \frac{q}{1-q}$ and transform (2) into

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & \sum_i L_\tau\left(t - y_i(w^T x_i + b)\right) \\ & \geq \sum_i L_\tau\left(1 - y_i(w^T x_i + b)\right), \forall t \in \mathbb{R}. \end{aligned}$$

The constraint is obviously non-convex for nonzero τ. We minimize $\sum_i L_\tau\left(1 - y_i(w^T x_i + b)\right)$ to approach the requirement, which results in the following SVM with pinball loss,

$$\min_{w,b} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^m L_\tau\left(1 - y_i(w^T x_i + b)\right). \tag{5}$$

We call (5) a pinball loss SVM (pin-SVM). As mentioned before, the proposed method preserves the elegance of the classical SVM: the only difference in form between the pin-SVM and the hinge loss SVM is that different losses are used.

Similarly to the hinge loss SVM, the pin-SVM can be extended to nonlinear classification, by introducing a nonlinear feature mapping $\phi(x)$ as follows,

$$\min_{w,b} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^m L_\tau\left(1 - y_i(w^T \phi(x_i) + b)\right).$$

The problem is further equivalently transformed into

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}w^T w + C\sum_{i=1}^m \xi_i \\ \text{s.t.} \quad & y_i\left[w^T\phi(x_i) + b\right] \geq 1 - \xi_i, i = 1,2,\ldots,m, \\ & y_i\left[w^T\phi(x_i) + b\right] \leq 1 + \frac{1}{\tau}\xi_i, i = 1,2,\ldots,m. \end{aligned} \tag{6}$$

Notice that when $\tau = 0$, the second constraint becomes $\xi_i \geq 0$ and (6) reduces to the hinge loss SVM.

B. Dual problem and kernel formulation

Now we introduce a kernel based formulation for the pinball loss SVM. The Lagrangian with $\alpha_i \geq 0, \beta_i \geq 0$ of (6) is

$$\begin{aligned} & \mathscr{L}(w,b,\xi;\alpha,\beta) \\ & = \tfrac{1}{2}w^T w + C\sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i\left(y_i\left[w^T\phi(x_i) + b\right] - 1 + \xi_i\right) \\ & \quad - \sum_{i=1}^m \beta_i\left(-y_i\left[w^T\phi(x_i) + b\right] + 1 + \tfrac{1}{\tau}\xi_i\right). \end{aligned}$$

According to

$$\begin{aligned} \frac{\partial\mathscr{L}}{\partial w} &= w - \sum_{i=1}^m(\alpha_i - \beta_i)y_i\phi(x_i) = 0, \\ \frac{\partial\mathscr{L}}{\partial b} &= -\sum_{i=1}^m(\alpha_i - \beta_i)y_i = 0, \\ \frac{\partial\mathscr{L}}{\partial \xi_i} &= C - \alpha_i - \frac{1}{\tau}\beta_i = 0, \forall i = 1,2,\ldots,m, \end{aligned}$$

the dual problem of (6) is obtained as follows,

$$\begin{aligned} \max_{\alpha,\beta} \quad & -\frac{1}{2}\sum_{i=1}^m\sum_{j=1}^m(\alpha_i - \beta_i)y_i\phi(x_i)^T\phi(x_j)y_j(\alpha_j - \beta_j) \\ & + \sum_{i=1}^m(\alpha_i - \beta_i) \\ \text{s.t.} \quad & \sum_{i=1}^m(\alpha_i - \beta_i)y_i = 0, \\ & \alpha_i + \frac{1}{\tau}\beta_i = C, i = 1,2,\ldots,m, \\ & \alpha_i \geq 0, \beta_i \geq 0, i = 1,2,\ldots,m. \end{aligned}$$

Introducing the positive definite kernel $K(x_i,x_j) = \phi(x_i)^T\phi(x_j)$ and variables $\lambda_i = \alpha_i - \beta_i$, we get

$$\begin{aligned} \max_{\lambda,\beta} \quad & -\frac{1}{2}\sum_{i=1}^m\sum_{j=1}^m \lambda_i y_i K(x_i,x_j)y_j\lambda_j + \sum_{i=1}^m \lambda_i \\ \text{s.t.} \quad & \sum_{i=1}^m \lambda_i y_i = 0, \\ & \lambda_i + (1 + \frac{1}{\tau})\beta_i = C, i = 1,2,\ldots,m, \\ & \lambda_i + \beta_i \geq 0, \beta_i \geq 0, i = 1,2,\ldots,m. \end{aligned} \tag{7}$$

Again, we observe the equivalence between the hinge loss SVM and the pin-SVM with $\tau = 0$: when τ is small enough, $(1 + \frac{1}{\tau})\beta_i$ can provide any positive value, thus the corresponding constraint is satisfied if and only if $0 \leq \lambda_i \leq C$. Hence, (7) reduces to the well-known dual formulation of the hinge loss SVM as follows,

$$\begin{aligned} \max_{\lambda} \quad & -\frac{1}{2}\sum_{i=1}^m\sum_{j=1}^m \lambda_i y_i K(x_i,x_j)y_j\lambda_j + \sum_{i=1}^m \lambda_i \\ \text{s.t.} \quad & \sum_{i=1}^m \lambda_i y_i = 0, \\ & 0 \leq \lambda_i \leq C, i = 1,2,\ldots,m. \end{aligned} \tag{8}$$

Denote the solution of (7) by λ^* and β^*. Then $\alpha^* = \lambda^* - \beta^*$ and we define the following set,

$$\mathcal{S} = \{i : \alpha_i^* \neq 0 \text{ and } \beta_i^* \neq 0\}.$$

According to the complementary slackness conditions, $\mathcal{S}$ defines the classification function $w^T\phi(x) + b$ by

$$y_i\left(w^T\phi(x_i) + b\right) = 1, \forall i \in \mathcal{S}.$$

This means that the elements of $\mathcal{S}$ play the role similar to the support vectors in the hinge loss SVM: $x_i, i \in \mathcal{S}$ determine the classification boundary. Fig.2 gives an intuitive example. We apply the hinge loss SVM and the pin-SVM ($\tau = 0.5$) with linear kernel to calculate classifiers for the data (both vertical and horizontal positions) shown in Fig.1(c). The obtained classification boundary and the hyperplanes equaling to ± 1 are shown in Fig.2, where the support vectors of the hinge loss SVM and the elements of $\mathcal{S}$ of the pin-SVM are marked by squares and circles, respectively.

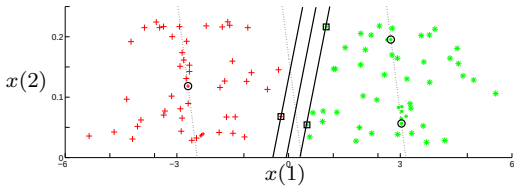


Fig. 2. Classification results for the data shown in Fig.1(c). For the result of the hinge loss SVM, the classification boundary with the hyperplanes equaling to ± 1 are shown by solid lines and the support vectors are marked by squares. For the result of the pin-SVM with $\tau = 0.5$, the classification boundary and the hyperplanes equaling to ± 1 are shown by dashed lines. The elements of $\mathcal{S}$ are marked by circles.

Therefore, similarly to the method of calculating the bias term for the hinge loss SVM, we can calculate the optimal b in the dual problem, denoted by b^*, from the following equations,

$$\sum_{i=1}^{m} \lambda_i^* y_j K(x_i, x_j) + b^* = 0, \forall j \in \mathcal{S}.$$

For each $x_j, j \in \mathcal{S}$, we calculate b^* by the above equation and use the average value as the result.

III. Properties of pinball loss for classification

A. Bayes rule

Binary classification problems have been widely investigated in statistical learning theory under the assumption that samples $\{x_i, y_i\}_{i=1}^m$ are independently drawn from a probability measure ρ. This probability measure is defined on $X \times Y$, where $X \subseteq \mathbb{R}^n$ is the input space and $Y = \{-1, 1\}$ represents two classes. The classification problem aims at producing a binary classifier $\mathcal{C} : X \to Y$ with a small misclassification error measured by

$$\mathcal{R}(\mathcal{C}) = \int_{X\times Y} \mathcal{I}_{y \neq \mathcal{C}(x)} d\rho = \int_X \rho(y \neq \mathcal{C}(x)|x) d\rho_X,$$

where $\mathcal{I}$ is the indicator function, ρ_X is the marginal distribution of ρ on X, and $\rho(y|x)$ is the conditional distribution of ρ at x. It should be pointed out that $\rho(y|x)$ is a binary distribution, which is given by $\mathrm{Prob}(y = -1|x)$ and $\mathrm{Prob}(y = 1|x)$. Define the Bayes classifier as

$$f_c(x) = \begin{cases} 1, & \text{if } \mathrm{Prob}(y = 1|x) \geq \mathrm{Prob}(y = -1|x), \\ -1, & \text{if } \mathrm{Prob}(y = 1|x) < \mathrm{Prob}(y = -1|x). \end{cases}$$

Then one can verify that f_c minimizes the misclassification error, i.e.,

$$f_c = \arg \min_{\mathcal{C}: X \to Y} \mathcal{R}(\mathcal{C}).$$

In practice, we are seeking a real-valued function $f : X \to \mathbb{R}$ and use its sign, i.e., $\mathrm{sgn}(f)$, to induce a binary classifier. In this case, the misclassification error becomes

$$\int_{X\times Y} \mathcal{I}_{y \neq \mathrm{sgn}(f)(x)} d\rho = \int_{X\times Y} L_{\mathrm{mis}}(yf(x)) d\rho,$$

where $L_{\mathrm{mis}}(u)$ is the misclassification loss defined as

$$L_{\mathrm{mis}}(u) = \begin{cases} 0, & u \geq 0, \\ 1, & u < 0. \end{cases}$$

Therefore, minimizing the misclassification error over real-valued functions will lead to a function, of which the sign is the Bayes classifier f_c. However, $L_{\mathrm{mis}}(u)$ is non-convex and discontinuous. To approach the misclassification loss, researchers have proposed some losses, shown in Fig.3. Fig.3(a) displays the hinge loss and the 2-norm loss, which are the most widely used losses for classification. To deal with outliers, the normalized sigmoid loss and the truncated hinge loss were introduced by [23] and [24], respectively and are shown in Fig.3(b). The robustness comes from the small deviation on the point away from the boundary, which results in the non-convexity. In this paper, we focus on insensitivity to noise around the decision boundary and improve the performance by giving penalty on $u > 0$, as illustrated in Fig.3(c). From this figure, one may find that the pinball loss is somehow strange that it gives penalty on the points which are classified correctly. In this section, we show that the pinball loss preserves good properties and then explain the reason of giving penalty on the correctly classified points. The first thing is that the pinball loss minimization also leads to the Bayes classifier. For any loss L, the expected *L-risk* of a measurable function $f : X \to \mathbb{R}$ is defined as follows,

$$\mathcal{R}_{L,\rho}(f) = \int_{X\times Y} L(1 - yf(x)) d\rho.$$

Minimizing the expected risk over all measurable functions results in function $f_{L,\rho}$, which is defined as follows,

$$f_{L,\rho}(x) = \arg\min_{t\in\mathbb{R}} \int_Y L\left(1 - y(x)t\right) d\rho(y|x), \ \forall x \in X.$$

Then for the pinball loss, we have the following theorem.

Theorem 1: Function $f_{L_\tau,\rho}$, which minimizes the expected L_τ-risk over all measurable functions $f : X \to Y$, is equal to the Bayes classifier, i.e., $f_{L_\tau,\rho}(x) = f_c(x), \forall x \in X$.

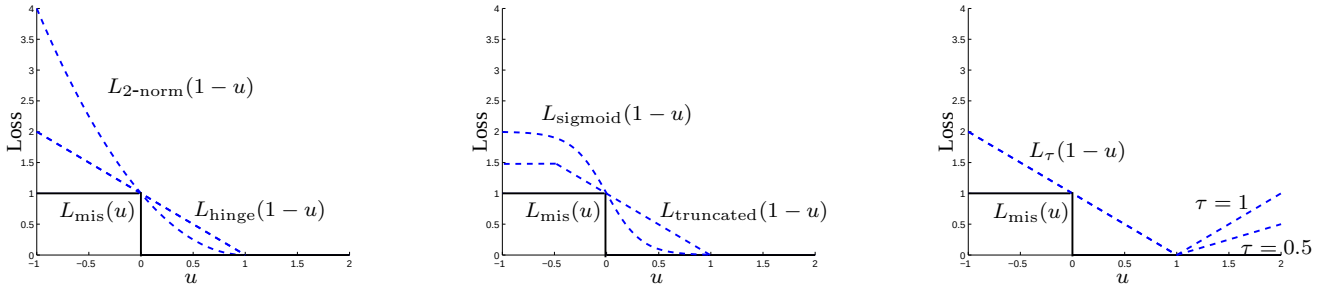
Proof: Simple calculation shows that

$$\begin{aligned} & \int_Y L_\tau\left(1 - y(x)t\right) d\rho(y|x) \\ = \ & L_\tau(1-t)\mathrm{Prob}(y = 1|x) + L_\tau(1+t)\mathrm{Prob}(y = -1|x) \\ = \ & \begin{cases} (1-t)\mathrm{Prob}(y = 1|x) - \tau(1+t)\mathrm{Prob}(y = -1|x), \\ \qquad\qquad t \leq -1, \\ (1-t)\mathrm{Prob}(y = 1|x) + (1+t)\mathrm{Prob}(y = -1|x), \\ \qquad\qquad -1 < t < 1, \\ \tau(t-1)\mathrm{Prob}(y = 1|x) + (1+t)\mathrm{Prob}(y = -1|x), \\ \qquad\qquad t \geq 1. \end{cases} \end{aligned}$$

Hence, when $\mathrm{Prob}(y = 1|x) > \mathrm{Prob}(y = -1|x)$, the minimal value is $2\mathrm{Prob}(y = -1|x)$, which is achieved by $t = 1$. When $\mathrm{Prob}(y = 1|x) < \mathrm{P}(y = -1|x)$, the minimal value is $2\mathrm{Prob}(y = 1|x)$, which is achieved by $t = -1$. When $\mathrm{Prob}(y = 1|x) = \mathrm{Prob}(y = -1|x)$, the minimal value is 1, which is achieved by any $t \in [-1, 1]$. Therefore, $f_{L_\tau,\rho}(x)$, which minimizes the expected risk measured by the pinball loss, has the following property,

$$f_{L_\tau,\rho}(x) = \begin{cases} 1, & \mathrm{Prob}(y = 1|x) \geq \mathrm{Prob}(y = -1|x), \\ -1, & \mathrm{Prob}(y = 1|x) < \mathrm{Prob}(y = -1|x), \end{cases}$$

that means $f_{L_\tau,\rho}(x) = f_c(x)$. ■



(a) (b) (c)

Fig. 3. The misclassification loss $L_{\text{mis}}(u)$ is shown by solid lines and some loss functions used for classification are displayed by dashed lines: (a) the hinge loss and the 2-norm loss [1]; (b) the normalized sigmoid loss [23] and the truncated hinge loss [24]; (c) the pinball loss with $\tau = 0.5$ and $\tau = 1$.

B. Bounding the misclassification error

From the fact that minimizing the pinball loss results in the Bayes classifier, one can see some rationality for using the pinball loss. In fact, the pinball loss meets the condition for margin-based losses, which requires that the loss is a function of yf ([25]). Moreover, a margin-based loss is called classification-calibrated in [26], if the minimizer of the related expected risk has the same sign as the Bayes rule for all $x : \rho(y = 1|x) \neq \frac{1}{2}$. According to Theorem 1, it can be verified that the pinball loss is classification-calibrated. Hence some important analysis on Fisher consistency and the risk bounds for classification-calibrated losses are valid for the pinball loss as well. But, similarly to the hinge loss, the pinball loss is neither a permissible surrogate [27] nor a proper loss [28] in classification problems.

In this subsection, we focus on the misclassification error for the pinball loss. In [21], one bound of the misclassification error has been given for any loss meeting the following conditions:

- $L(1-u)$ is convex with respect to u;
- $L(1-u)$ is differentiable at $u = 0$ and $\frac{dL(1-u)}{du}\big|_{u=0} < 0$;
- $\min\{u : L(1-u) = 0\} = 1$;
- $\frac{d^2L(1-u)}{du^2}\big|_{u=1} > 0$.

If these conditions are satisfied, then there exists a constant c_L such that for any measurable function $f : X \to \mathbb{R}$,

$$\mathcal{R}_{L_{\text{mis}},\rho}(\text{sgn}(f)) - \mathcal{R}_{L_{\text{mis}},\rho}(f_c) \leq c_L \sqrt{\mathcal{R}_{L,\rho}(f) - \mathcal{R}_{L,\rho}(f_{L,\rho})}. \tag{9}$$

For details, please refer to Theorem 10.5 in [21]. The property holds for q-norm loss ($q \geq 2$), ℓ_2 loss and so on. The inequality (9) plays an essential role in error analysis of classification algorithms associated with loss L. Concretely, we denote $f_{L,\mathbf{z}}$ as the output function of the concerned classification algorithm based on loss L and samples $\mathbf{z}$. As the minimal classification error is given by $\mathcal{R}_{L_{\text{mis}},\rho}(f_c)$, the performance of the algorithm can be evaluated by $\mathcal{R}_{L_{\text{mis}},\rho}(\text{sgn}(f_{L,\mathbf{z}})) - \mathcal{R}_{L_{\text{mis}},\rho}(f_c)$, which can be further estimated by bounding $\mathcal{R}_{L,\rho}(f_{L,\mathbf{z}}) - \mathcal{R}_{L,\rho}(f_{L,\rho})$ based on (9). Under the i.i.d. assumption for sampling, one may expect that $\mathcal{R}_{L,\rho}(f_{L,\mathbf{z}}) - \mathcal{R}_{L,\rho}(f_{L,\rho})$ will tend to zero in probability as the sample size increases. The convergence behavior of $\mathcal{R}_{L,\rho}(f_{L,\mathbf{z}}) - \mathcal{R}_{L,\rho}(f_{L,\rho})$ has been extensively studied in the literatures, e.g., [21] and [22].

For the hinge loss, there is a tighter bound on the misclassification error. The following bound was given in [29] and is known as Zhang's inequality,

$$\mathcal{R}_{L_{\text{mis}},\rho}(\text{sgn}(f)) - \mathcal{R}_{L_{\text{mis}},\rho}(f_c) \leq \mathcal{R}_{L_{\text{hinge}},\rho}(f) - \mathcal{R}_{L_{\text{hinge}},\rho}(f_c).$$

According to the facts that

$$\mathcal{R}_{L_\tau,\rho}(f) \geq \mathcal{R}_{L_{\text{hinge}},\rho}(f), \forall f,$$

and

$$\mathcal{R}_{L_\tau,\rho}(f_c) = \mathcal{R}_{L_{\text{hinge}},\rho}(f_c),$$

we can bound the classification error for the pinball loss, represented in the following theorem,

Theorem 2: For any probability measure ρ and any measurable function $f : X \to \mathbb{R}$,

$$\mathcal{R}_{L_{\text{mis}},\rho}(\text{sgn}(f)) - \mathcal{R}_{L_{\text{mis}},\rho}(f_c) \leq \mathcal{R}_{L_\tau,\rho}(f) - \mathcal{R}_{L_\tau,\rho}(f_c). \tag{10}$$

The improvement of Theorem 2 from (9) arises in two aspects. First, a bound tighter than (9) can be given; second, not like (9), the right hand side of (10) is directly related to the Bayes classifier, since we have $f_{L_\tau,\rho}(x) = f_c(x)$ as proved in Theorem 1.

C. Noise insensitivity

In the previous sections, we have shown that minimizing the risk of the pinball loss leads to the Bayes classifier and the classification error bound for the pinball loss is the same as that for the hinge loss. However, using the pinball loss instead of the hinge loss will result in losing the sparsity. The technique for enhancing sparsity of the pin-SVM will be discussed in Section IV. In this subsection, we explain the benefit of giving penalty on correctly classified points. The main benefit is that the pinball loss minimization enjoys insensitivity with respect to noise around the decision boundary.

For easy comprehension, we focus on a linear classifier. Define the generalized sign function $\text{sgn}_\tau(u)$ as

$$\text{sgn}_\tau(u) = \begin{cases} 1, & u > 0, \\ [-\tau, 1], & u = 0, \\ -\tau, & u < 0. \end{cases}$$

$\mathrm{sgn}_\tau(u)$ is the subgradient of the pinball loss $L_\tau(u)$ and then the optimality condition for (5) can be written as

$$\mathbf{0} \in \frac{w}{C} - \sum_{i=1}^{m} \mathrm{sgn}_\tau(1 - y_i(w^T x_i + b)) y_i x_i,$$

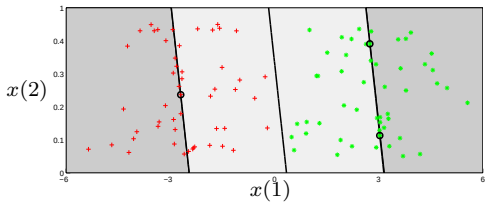
where $\mathbf{0}$ denotes the vector of which all the components equal zero. For given w, b, the index set is partitioned into three sets,

$$\begin{aligned} \mathcal{S}_+^{w,b} &= \left\{ i : 1 - y_i(w^T x_i + b) > 0 \right\}, \\ \mathcal{S}_-^{w,b} &= \left\{ i : 1 - y_i(w^T x_i + b) < 0 \right\}, \\ \mathcal{S}_0^{w,b} &= \left\{ i : 1 - y_i(w^T x_i + b) = 0 \right\}. \end{aligned}$$

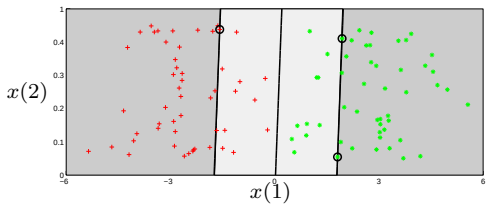
Use the notations $\mathcal{S}_+^{w,b}, \mathcal{S}_-^{w,b}, \mathcal{S}_0^{w,b}$, the optimality condition can be written as the existence of $\zeta_i \in [-\tau, 1]$ such that

$$\frac{w}{C} - \sum_{i \in \mathcal{S}_+^{w,b}} y_i x_i + \tau \sum_{i \in \mathcal{S}_-^{w,b}} y_i x_i - \sum_{i \in \mathcal{S}_0^{w,b}} \zeta_i y_i x_i = \mathbf{0}. \quad (11)$$

The above condition shows that τ controls the numbers of points in $\mathcal{S}_-^{w,b}$ and $\mathcal{S}_+^{w,b}$. When $\tau = 1$, both sets contain many points and hence the result is less sensitive to zero-mean noise on feature. When τ is small, there are few points in $\mathcal{S}_+^{w,b}$ and the result is sensitive. Consider again the data shown in Fig.1(b). We use the pin-SVM with $\tau = 0.5$ and illustrate the result in Fig.4(a), where $x_i, i \in \mathcal{S}_0^{w,b}$ are marked by circles, the region of $x_i, i \in S_-^{w,b}$ is shown shaded and the region of $x_i, i \in S_+^{w,b}$ is shown lightly shaded. Since there are plenty of points in the lightly shaded region, the sum of $x_i, i \in S_+^{w,b}$ is insensitive to noise on x_i.



(a)



(b)

Fig. 4. Classification results for data in Fig.1(b). The points in $\mathcal{S}_0^{w,b}$ are marked by circles, the regions corresponding to $\mathcal{S}_-^{w,b}$ and $\mathcal{S}_+^{w,b}$ are shown shaded and lightly shaded, respectively. (a) the pin-SVM with $\tau = 0.5$; (b) the pin-SVM with $\tau = 0.1$.

Along with the decrease of τ, the number of elements in $\mathcal{S}_+^{w,b}$ is becoming smaller. As an example, Fig.4(b) illustrates the corresponding regions for the pin-SVM with $\tau = 0.1$. When $\tau = 0$, the pin-SVM reduces to the hinge loss SVM and there is no point or only a small number of points in $\mathcal{S}_+^{w,b}$. Therefore, the feature noise around the decision boundary will significantly affect the classification result. To make a comparison, consider the following example. The input data are uniformly located in the domain $\{x : 0 \le x(1) \le 1, 0 \le x(2) \le 1\}$ and the boundary of the two classes is $4(x(1) - 0.5)^3 - x(2) + 0.5 = 0$. The boundary is illustrated by dashed lines in Fig.5(a) and Fig.5(b) and the values of $4(x(1) - 0.5)^3 - x(2) + 0.5$ are displayed by different colors. We first use input data shown in Fig.5(a) and Fig.5(b), where data in two classes are marked by green stars and red crosses, respectively. The hinge loss SVM (8) and the pin-SVM (7) are applied to establish a nonlinear classifier. In this study, the RBF kernel $K_\sigma(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2/\sigma^2\right)$ with $\sigma = 0.5$ is used and C is set to be 1000. As the results showing, the classification performance of the hinge loss SVM and the pin-SVM are both satisfactory. Next, we add noise on the features and the noise follows the uniform distribution on $[-0.2, 0.2]$. Then the hinge loss SVM (8) and the pin-SVM (7) are used again to do classification. The obtained classifiers are illustrated in Fig.5(c) and Fig.5(d), which show that the result of the pin-SVM is less sensitive than that of the hinge loss SVM.

D. Scatter minimization

The mechanism of the pin-SVM can be interpreted from scatter minimization as well. Points in $\mathcal{S}_0^{w,b}$ determine two hyperplanes $\mathcal{H}_\mathbf{I} : \{x : w^T x + b = 1\}$ and $\mathcal{H}_\mathbf{II} : \{x : w^T x + b = -1\}$ and $\|w\|^2$ corresponds to the distance between them. We can use the sum of the distances to one given point to measure the scatter. In the projected space related to w, the scatter of $x_i, i \in \mathbf{I}$ around point x_{i_0} can be defined as

$$\sum\nolimits_{i \in \mathbf{I}} \left| w^T (x_{i_0} - x_i) \right|.$$

If $i_0 \in \mathcal{S}_0^{w,b} \bigcap \mathbf{I}$, i.e., $w^T x_{i_0} + b = 1, y_{i_0} = 1$, then

$$\sum\nolimits_{i \in \mathbf{I}} \left| w^T (x_{i_0} - x_i) \right| = \sum\nolimits_{i \in \mathbf{I}} \left| 1 - y_i(w^T x_i + b) \right|.$$

A similar analysis holds for $x_i, i \in \mathbf{II}$. Therefore,

$$\min_{w,b} \quad \frac{1}{2}\|w\|^2 + C_1 \sum\nolimits_{i=1}^{m} \left| 1 - y_i(w^T x_i + b) \right| \quad (12)$$

can be interpreted as to maximize the distance between hyperplanes $\mathcal{H}_\mathbf{I}$ and $\mathcal{H}_\mathbf{II}$ and meanwhile to minimize the scatters around them. The above argument can be discussed under the framework of Fisher discriminant analysis ([30], [19]). Similar analysis exists for ℓ_2 loss, which was proposed by [31] and gives penalty on correctly classified points as well. One can refer to [32] for the Fisher discriminant analysis on ℓ_2 loss, for which the sum of the squared distance from the class center is used to measure the scatter.

In the pin-SVM (5), the absolute value used in (12) is extended to L_τ. The pinball loss minimization can be regarded as that we consider the within-class scatter and the misclassification error together. The pin-SVM (5) is then interpreted as a trade-off between small scatter and small misclassification: introducing the misclassification term $C_2 L_{\mathrm{hinge}}(1 - y_i(w^T x_i + b))$ into (12), we obtain the pin-SVM (5) with $C = C_1 + C_2$ and $\tau = \frac{C_1}{C}$. This interpretation tells us that the reasonable range of τ is $0 \le \tau \le 1$.

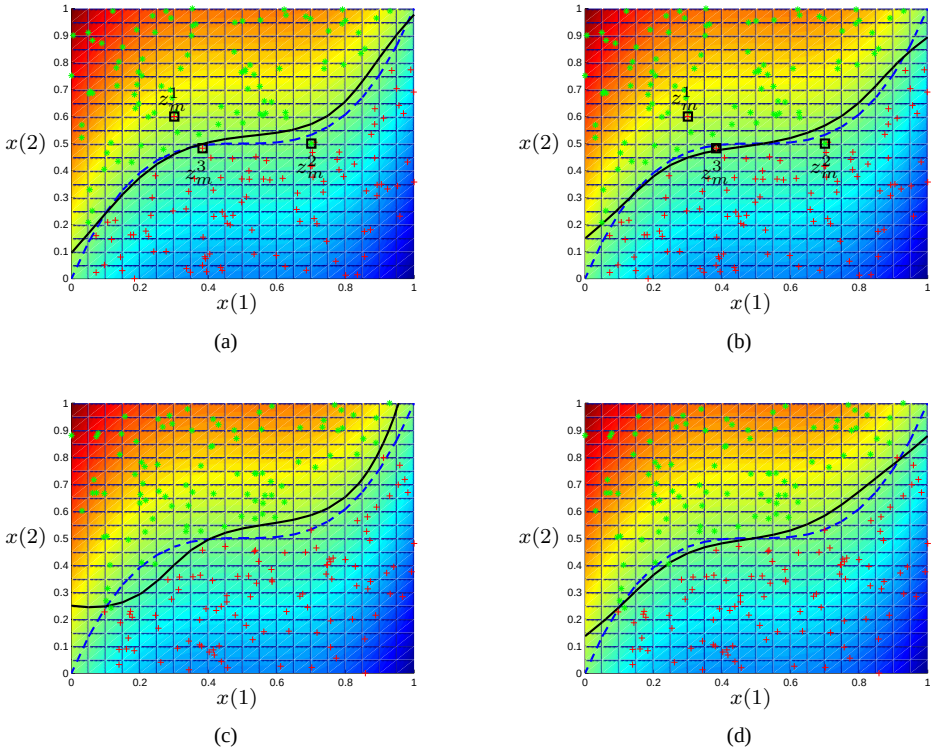


Fig. 5. Data of two classes are marked by green stars and red crosses. The dashed lines illustrate the boundary. The input data of (c) (d) are generated by adding noise on the positions. In (a) and (c), the boundaries obtained by the hinge loss SVM are shown by solid lines; In (b) and (d) the boundaries obtained by the pin-SVM with $\tau = 0.5$ are shown by solid lines. The comparison shows that the pin-SVM is less sensitive to noise than the hinge loss SVM. In (a) (b), there are 3 squares, indicating the additional data, which are used to compute the sensitivity curves shown in Fig.7.

Small within-class scatter and small misclassification error are two desirable targets for a good classifier. The hinge loss puts emphasis on misclassification error, the absolute loss puts emphasis on within-class scatter, and the pinball loss is a trade-off considering the two targets together. In the following two-dimensional classification task, the data come from two Gaussian distributions. The p.d.f. of the two distributions are shown in Fig.6(a). For this problem, the hinge loss SVM, which maximizes the shortest distance between two classes, gives a very good classifier, displayed by the solid lines. However, minimizing the within-class scatter defines the horizontal axis as the classification boundary, which is certainly very bad. The reason is that scatter measured by sum of the absolute divergence lacks of invariance for scaling. Therefore, normalization technique is required for pre-processing. We simply scale each feature such that all the features have the same range or the same variance. In this example, feature $x(1)$ can distinguish the two classes while feature $x(2)$ for the two classes are the same. Hence, when the ranges or the variances of $x(1)$ and $x(2)$ are equal, one can expect that the within-class scatter in $x(1)$ is smaller than that in $x(2)$, because the margin between classes in $x(1)$ is larger.

In Fig.6(b), the normalized distributions (the range of each feature is $[-1, 1]$) are shown. Clearly, minimizing misclassification error and minimizing small within-class scatter both give satisfactory results, if there are plenty of training data. We randomly sample 50 points from each class. For this training trial, the hinge loss SVM uses the three nearest points to determine a classifier, which differs from the ideal one. In contrast, the result of the pin-SVM is more stable for re-sampling.

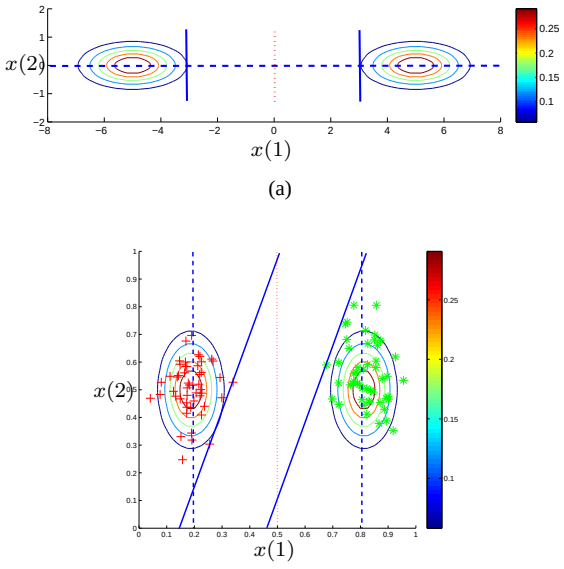


Fig. 6. Contour maps of p.d.f., the ideal decision boundary (red dotted lines), the hyperplanes minimizing the misclassification error (blue solid lines), and the hyperplanes minimizing the within-class scatter (blue dashed lines). (a) original problem; (b) normalized problem and one sampling set.

E. Discussion about robustness

In the previous discussion, the pinball loss SVM has shown noise insensitivity and re-sampling stability. In this subsection, we analyze the robustness to outliers by considering the perturbation on the probability measure. We denote the result obtained from SVM on the distribution ρ as $\hat{f}_{L,\rho}$, where L can be any loss function, such as L_{hinge} and L_τ. Suppose ρ is corrupted by a distribution $\tilde{\rho}$ defined on $X \times Y$ to be $(1-\theta)\rho + \theta\tilde{\rho}$, where $0 < \theta < 1$. Then the impact of the distribution on the classification result can be measured by

$$\left\| \frac{\hat{f}_{L,(1-\theta)\rho+\theta\tilde{\rho}} - \hat{f}_{L,\rho}}{\theta} \right\|_{H_K}, \tag{13}$$

where H_K is a reproducing kernel Hilbert space of a given kernel K. When there is no bias, the upper bound of (13) has been given in [20]. Now we restate the result as follows: for any continuous and convex loss function L, and a bounded continuous kernel K, there is a positive constant h_L, which is proportional to the Lipschitz constant of L, such that

$$\left\| \frac{\hat{f}_{L,(1-\theta)\rho+\theta\tilde{\rho}} - \hat{f}_{L,\rho}}{\theta} \right\|_{H_K} \leq h_L \|\rho - \tilde{\rho}\|_{\mathcal{M}}, \forall \theta > 0, \tag{14}$$

where $\|\cdot\|_{\mathcal{M}}$ is the norm of total variation of a signed measure. The definition of this norm can be found in [20] and references therein. According to (14), we know the robustness of the hinge loss SVM and the pin-SVM, since L_{hinge} and L_τ are (globally) Lipschitz continuous. One noticeable point is that though the hinge loss SVM is robust to outliers, it is sensitive to noise around the boundary. To further improve the robustness to outliers of the pin-SVM, the truncating technique used for the hinge loss in [24] is a potential direction.

Though it is hard to theoretically compare the robustness of the hinge loss SVM and the pin-SVM, we can use a sensitivity curve to evaluate the performance numerically. The sensitivity curve can be interpreted as a finite sample version of the influence function, which is a popular tool in robustness analysis and the definition can be found in Definition 10.4 in [22]. Denote the function obtained from SVM with loss L and input data $\mathbf{z} = \{x_i, y_i\}_{i=1}^{m-1}$ as $\hat{f}_{L,\mathbf{z}}$. The sensitivity curve at an additional point $z_m = \{x_m, y_m\}$ then is defined as

$$SC(z_m; \hat{f}_{L,\mathbf{z}}) = m\big(\hat{f}_{L,(\mathbf{z},z_m)} - \hat{f}_{L,\mathbf{z}}\big).$$

Consider the data shown in Fig.5(a) and Fig.5(b). Three points marked by squares are added into training set in turn and $SC(z_m; \hat{f}_{L_\tau,\mathbf{z}})$, $SC(z_m; \hat{f}_{L_{\text{hinge}},\mathbf{z}})$ are calculated. The additional data $z_m = \{x_m, y_m\}$ are listed below: the first additional data is $z_m^1 = \{[0.3, 0.6], -1\}$, which is away from the classification boundary and has a wrong label; the second data is $z_m^2 = \{[0.7, 0.5], 1\}$, which has a wrong label as well and is near the boundary; the third data $z_m^3 = \{[0.38, 0.48], -1\}$ has a right label and is near the boundary. With each additional point, we use the hinge loss SVM and the pin-SVM to do classification. The results are compared with the classifiers obtained from the data without additional points and the sensitivity curves are illustrated in Fig.7(a) – Fig.7(f). These figures shows that $SC(z_m; \hat{f}_{L_\tau,\mathbf{z}})$ is significantly smaller than $SC(z_m; \hat{f}_{L_{\text{hinge}},\mathbf{z}})$, indicating the robustness of the pin-SVM.

IV. Pinball loss with ε insensitive zone

A. Sparsity and ε insensitive zone

One good property of the hinge loss SVM is that the number of nonzero dual variables is small, i.e., the solution of the hinge loss SVM (8) is sparse. Based on the complementary slackness conditions and the constraint of dual problem (7), we know

$$y_i(w^T\phi(x) + b) > 1 + \xi_i \quad \Rightarrow \quad \alpha_i = 0, \beta_i = \tau C,$$

and

$$y_i(w^T\phi(x) + b) < 1 - \frac{1}{\tau}\xi_i \quad \Rightarrow \quad \alpha_i = C, \beta_i = 0.$$

Therefore, $\lambda_i \neq 0$ for most i, which means that the pinball loss SVM (6) loses the sparsity of the hinge loss SVM.

To achieve sparsity, we introduce an insensitive zone to the pin-SVM, then any point located in the insensitive zone corresponds to a zero dual variable. For this purpose, we extend the pinball loss $L_\tau(u)$ to the following loss with ε insensitive zone,

$$L_\tau^\varepsilon(u) = \begin{cases} u - \varepsilon, & u \geq \varepsilon, \\ 0, & -\varepsilon \leq u \leq \varepsilon, \\ -\tau(u + \varepsilon), & u < -\varepsilon, \end{cases}$$

where $\varepsilon \geq 0$. When $\varepsilon = 0$, the above loss reduces to the pinball loss. The lengthes of insensitive zones for $u > 0$ and $u < 0$ can be different. $L_\tau(u)$ is related to $\frac{\tau}{1+\tau}$-quantile value of each class and the ratio of the lengths of region corresponding to $\mathcal{S}_+^{w,b}$ and $\mathcal{S}_+^{w,b}$ is related to τ, which can be seen from Fig.4. Therefore, it is reasonable that the lengths of insensitive zones are related with τ. Specifically, the smaller τ is, the larger the insensitive zone for $u < 0$ should be. Then in this paper, we redefine the pinball loss with ε insensitive zone as follows,

$$L_\tau^\varepsilon(u) = \begin{cases} u - \varepsilon, & u > \varepsilon, \\ 0, & -\frac{\varepsilon}{\tau} \leq u \leq \varepsilon, \\ -\tau(u + \frac{\varepsilon}{\tau}), & u < -\frac{\varepsilon}{\tau}. \end{cases}$$

In Fig.8, $L_\tau^\varepsilon(u)$ with $\tau = 0.5, \varepsilon = 0.2$ is illustrated.

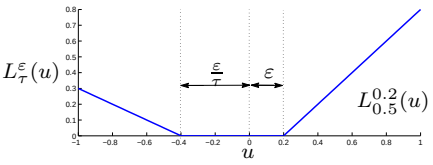


Fig. 8. An example of a pinball loss with insensitive zone $L_\tau^\varepsilon(u)$.

Applying $L_\tau^\varepsilon(u)$ as the loss function, we get the following formulation, called ε insensitive zone pinball loss SVM,

$$\min_{w,b} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} L_\tau^\varepsilon\left(1 - y_i(w^T\phi(x_i) + b)\right), \tag{15}$$

which can be equivalently written as,

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}w^T w + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y_i\left[w^T\phi(x_k) + b\right] \geq 1 - (\xi_i + \varepsilon), \\ & y_i\left[w^T\phi(x_i) + b\right] \leq 1 + \frac{1}{\tau}(\xi_i + \varepsilon), \\ & \xi_i \geq 0, \quad i = 1, 2, \ldots, m. \end{aligned} \tag{16}$$

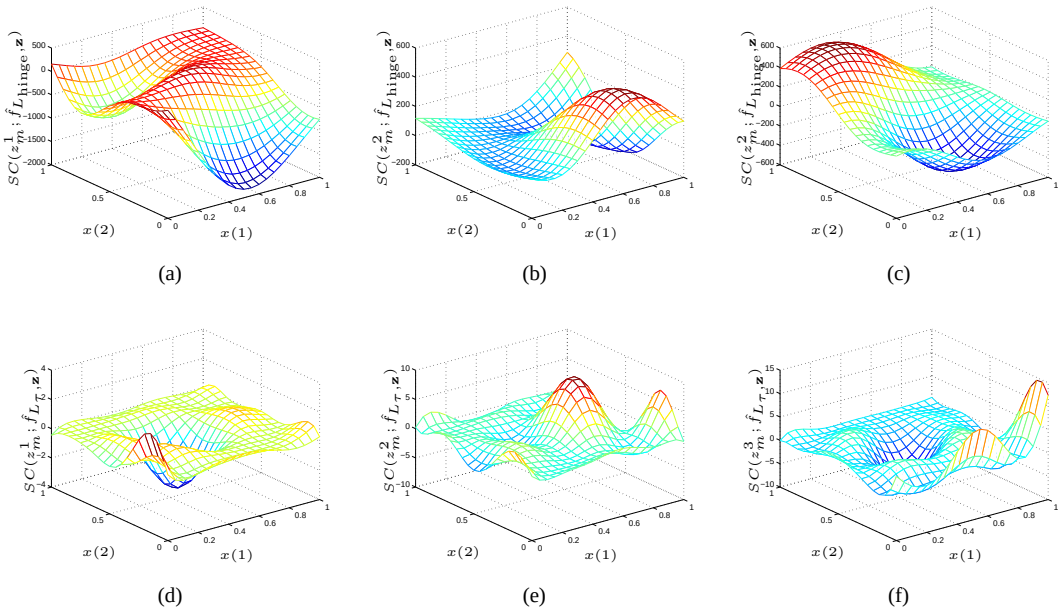


Fig. 7. The sensitivity curves of the hinge loss SVM and the pinball loss SVM. Data of two classes are marked by green stars and red crosses in Fig.5, where the additional data are marked by squares: (a) (b) (c) show the sensitivity curve of the hinge loss SVM for additional data $z_m^1 = \{[0.3, 0.6], -1\}$, $z_m^2 = \{[0.7, 0.5], 1\}$, and $z_m^3 = \{[0.38, 0.48], -1\}$, respectively; (d) (e) (f) show the sensitivity curve of the pin-SVM with $\tau = 0.5$ for the same additional data corresponding to (a) (b) (c), respectively.

B. Dual problem

Following the same way of introducing positive definite kernel for (6), we consider the dual formulation of (16) and use the kernel trick to get a nonlinear classifier. The Lagrangian with $\alpha_i \geq 0, \beta_i \geq 0$ of (16) is

$$\begin{aligned} & \mathscr{L}(w, b, \xi; \alpha, \beta, \gamma) \\ = \; & \tfrac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \gamma_i \xi_i \\ & - \sum_{i=1}^{m} \alpha_i \left(y_i \left[w^T \phi(x_k) + b \right] - 1 + \xi_i + \varepsilon \right) \\ & - \sum_{i=1}^{m} \beta_i \left(-y_i \left[w^T \phi(x_i) + b \right] + 1 + \tfrac{1}{\tau} \xi_i + \tfrac{1}{\tau} \varepsilon \right). \end{aligned}$$

Then according to

$$\begin{aligned} \tfrac{\partial \mathscr{L}}{\partial w} &= w - \sum_{i=1}^{m} (\alpha_i - \beta_i) y_i \phi(x_i) = 0, \\ \tfrac{\partial \mathscr{L}}{\partial b} &= - \sum_{i=1}^{m} (\alpha_i - \beta_i) y_i = 0, \\ \tfrac{\partial \mathscr{L}}{\partial \xi_i} &= C - \alpha_i - \tfrac{1}{\tau} \beta_i - \gamma_i = 0, \forall i = 1, 2, \ldots, m, \end{aligned}$$

we get the dual problem of (16) as follows,

$$\begin{aligned} \max_{\alpha, \beta, \gamma} \quad & -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} (\alpha_i - \beta_i) y_i \phi(x_i)^T \phi(x_j) y_j (\alpha_j - \beta_j) \\ & + \sum_{i=1}^{m} (\alpha_i - \beta_i - \varepsilon \gamma_i) \\ \text{s.t.} \quad & \sum\nolimits_{i=1}^{m} (\alpha_i - \beta_i) y_i = 0, \\ & \alpha_i + \frac{1}{\tau} \beta_i - \gamma_i = C, i = 1, 2, \ldots, m, \\ & \alpha_i \geq 0, \beta_i \geq 0, \gamma_i \geq 0, i = 1, 2, \ldots, m. \end{aligned} \tag{17}$$

Again, we introduce the positive definite kernel $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$, variables $\lambda_i = \alpha_i - \beta_i$ and get

$$\begin{aligned} \max_{\lambda, \beta, \gamma} \quad & -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i y_i K(x_i, x_j) y_j \lambda_j + \sum_{i=1}^{m} (\lambda_i + \varepsilon \gamma_i) \\ \text{s.t.} \quad & \sum\nolimits_{i=1}^{m} \lambda_i y_i = 0, \\ & \lambda_i + (1 + \frac{1}{\tau}) \beta_i + \gamma_i = C, i = 1, 2, \ldots, m, \\ & \lambda_i + \beta_i \geq 0, \beta_i \geq 0, \gamma_i \geq 0, i = 1, 2, \ldots, m. \end{aligned} \tag{18}$$

Using the ε insensitive zone pin-SVM (18), we can get the result with sparsity, that means there are some i satisfying $\lambda_i = 0$. Meanwhile, the ε insensitive zone pin-SVM is robust as well, since the Lipschitz constant of L_τ^ε is the same as that of L_τ. However, $L_\tau^\varepsilon(u)$ loses other properties of the pinball loss. For example, minimizing $L_\tau^\varepsilon(u)$ will not lead to the Bayes classifier, unless $\varepsilon = 0$. The corresponding misclassification error cannot be bounded by (10) neither. To discuss noise insensitivity, we consider the points in $\mathcal{S}_+^{w,b}$, which are partitioned into two parts: $\mathcal{S}_{++}^{w,b} = \{i, 1 - y_i(w^T x + b) > \varepsilon\}$ and $\mathcal{S}_{+-}^{w,b} = \{i, 0 < 1 - y_i(w^T x + b) < \varepsilon\}$. The optimality condition of the ε insensitive zone pin-SVM is similar to (11), but the term about $\mathcal{S}_+^{w,b}$ turns to be $\mathcal{S}_{++}^{w,b}$. Compared to the pin-SVM, the ε insensitive zone pin-SVM is more sensitive to the feature noise around the decision boundary, since the number of elements in $\mathcal{S}_{++}^{w,b}$ is less than that in $\mathcal{S}_+^{w,b}$.

V. Numerical Experiments

Before discussing numerical experiments, we first explain some issues about solving the pinball loss SVM. Like the hinge loss SVM, the pin-SVM is a linearly constrained quadratic programming (QP). In the primal space, there are

the same number of variables involved in the pin-SVM and the hinge loss SVM. The numbers of constraints are the same as well. In the dual formulation, the pin-SVM (7) has $2m$ variables, $m+1$ equality constraints, and $2m$ inequality constraints. Meanwhile, in (8), there are m variables, one equality constraint, and $2m$ inequality constraints. Hence, the computational complexity of the nonparametric pin-SVM is approximately the same as that of the hinge loss SVM. In this paper, we simply use a QP solver embedded in Matlab to solve the pin-SVM but leave the study on efficient algorithms for further work. In the following experiments, the data are normalized before training. In the following, the hinge loss SVM is written as C-SVM for short.

A. Synthetic data with noise

The purpose of proposing the pin-SVM is to deal with noise around the decision boundary. To illustrate its performance, we first consider a two-dimensional example, for which the samples come from two Gaussian distributions with equal probability: $x_i, i \in \mathbf{I} \sim \mathcal{N}(\mu_1, \Sigma_1)$, $x_i, i \in \mathbf{II} \sim \mathcal{N}(\mu_2, \Sigma_2)$, where $\mu_1 = [0.5, -3]^T, \mu_2 = [-0.5, 3]$ and

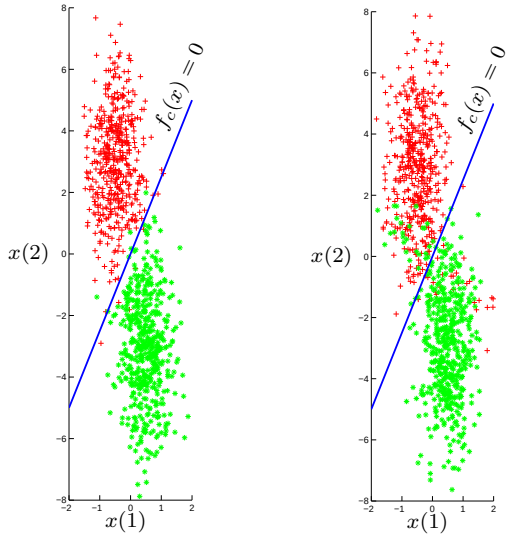
$$\Sigma_1 = \Sigma_2 = \begin{bmatrix} 0.2 & 0 \\ 0 & 3 \end{bmatrix}.$$

For this experiment, the Bayes classifier is $f_c(x) = 2.5x(1) - x(2)$. We display the Bayes classifier and one training set with 500 samplings in Fig.9(a). We generate m ($= 50, 100, 200, 500$) data, and apply the linear C-SVM and the pin-SVM to calculate the classification boundary $x(2) = w(1)x(1) + b$. The decision boundary given by the Bayes classifier is $x(2) = 2.5x(1)$. The ideal result is $w(1) = 2.5$ and $b = 0$. We repeat the sampling and training process 100 times, then report the mean and the standard deviation in Table I. Both the pin-SVM and the C-SVM will converge to the Bayes classifier and hence the average values are quite good. However, the deviation of the results of the C-SVM is significantly larger than that of the pin-SVM. This observation means that the pin-SVM is more stable for re-sampling, which implies the potential advantage of the pin-SVM for large scale problems. For example, a preliminary study [33] shows that the pin-SVM has a better convergence behavior on stochastic gradient method than the C-SVM.

Now we add noise into the training set. The labels of the noise points are selected from $\{1, -1\}$ with equal probability. The positions of these points follow Gaussian distribution $\mathcal{N}(\mu_n, \Sigma_n)$ with $\mu_n = [0, 0]^T$ and

$$\Sigma_n = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}.$$

This noise affects the labels around the boundary and the level of the noise is controlled by the ratio of the noise data in the training set, denoted by ξ. In Fig.9(b), one training set with $\xi = 0.1$ is shown. Such noise will not change the Bayes classifier, but it can affect the result of SVMs, as shown in Table II, which gives the mean and the standard deviation for repeating the above process 100 times. In this experiment, the classifier obtained by the C-SVM is quite different from



(a) (b)

Fig. 9. The points in two classes are shown by red crosses and green stars respectively. The blue solid lines give the Bayes classification boundary. (a) Data 1; (b) Data 1 with noise.

the Bayes classifier, implying the sensitivity of the C-SVM to noise around the boundary. One efficient way to improve the performance of SVM for noisy input is to allow small changes in the training features. This method was proposed in [2] and called total support vector classification (TSVC), which can be implemented by iteratively solving a series of C-SVMs. The computational time for the TSVC is larger than that of the C-SVM and it can improve the performance for this experiment. In contrast, the pin-SVM can give a better result and the computational time is similar to that of the C-SVM, as analyzed before.

In the above, the pin-SVM shows its insensitivity for noise around the decision boundary. In the following, we consider feature noise used in [2], where the training data are corrupted by zero-mean Gaussian noise. The covariance matrix is $\sigma_i \mathbf{E}$, where $\mathbf{E}$ is an identity matrix and σ_i is randomly selected from $[0.1, 0.8]$ with uniform distribution. Outlier effect is also added by randomly choosing $0.1m$ data and corrupting their features by noise with σ_i taken from $[0.5, 2]$. We use the pin-SVM, the C-SVM, and the TSVC to deal with the training data corrupted by noise. Then 5000 testing data are generated and the average classification accuracy for 20 trials is reported in the first and the second line of Table III.

TABLE III
CLASSIFICATION ACCURACY FOR NOISE PROVIDED IN [2]

	m	pin-SVM $\tau = 1$	0.5	0.2	0.1	C-SVM	TSVC
Data 1	50	97.64	97.69	97.73	97.66	96.55	97.05
	100	97.86	97.85	97.80	97.77	96.83	97.46
Data 2	50	92.04	92.02	93.19	93.82	94.45	95.46
	100	93.15	93.40	95.38	96.09	96.15	96.93

TABLE I
CLASSIFICATION BOUNDARY FOR DATA 1

Method		$m = 50$	$m = 100$	$m = 200$	$m = 500$
pin-SVM	$w(1)$	2.473 ± 1.112	2.550 ± 0.650	2.543 ± 0.536	2.585 ± 0.255
$\tau = 1$	b	0.001 ± 0.394	-0.005 ± 0.282	0.000 ± 0.190	0.001 ± 0.099
pin-SVM	$w(1)$	2.488 ± 1.035	2.582 ± 0.785	2.565 ± 0.611	2.577 ± 0.336
$\tau = 0.5$	b	-0.002 ± 0.364	0.025 ± 0.243	0.007 ± 0.204	0.008 ± 0.134
pin-SVM	$w(1)$	2.552 ± 1.018	2.632 ± 0.707	2.588 ± 0.577	2.600 ± 0.338
$\tau = 0.2$	b	0.048 ± 0.376	-0.043 ± 0.269	-0.017 ± 0.213	-0.001 ± 0.111
pin-SVM	$w(1)$	2.697 ± 1.228	2.649 ± 0.772	2.587 ± 0.589	2.539 ± 0.408
$\tau = 0.1$	b	0.053 ± 0.402	-0.024 ± 0.275	-0.009 ± 0.235	-0.005 ± 0.133
C-SVM	$w(1)$	2.431 ± 1.816	2.555 ± 1.313	2.686 ± 1.026	2.380 ± 0.556
	b	0.034 ± 0.789	0.030 ± 0.483	-0.011 ± 0.337	-0.002 ± 0.190

TABLE II
CLASSIFICATION BOUNDARY FOR NOISE CORRUPTED DATA 1

Method		$m = 100, \xi = 0.05$	$m = 100, \xi = 0.10$	$m = 200, \xi = 0.05$	$m = 200, \xi = 0.10$
pin-SVM	$w(1)$	2.400 ± 0.787	2.363 ± 0.869	2.356 ± 0.516	2.332 ± 0.578
$\tau = 1$	b	-0.064 ± 0.259	0.010 ± 0.262	0.044 ± 0.196	0.016 ± 0.169
pin-SVM	$w(1)$	2.324 ± 0.708	2.244 ± 0.776	2.358 ± 0.587	2.280 ± 0.525
$\tau = 0.5$	b	-0.038 ± 0.230	-0.013 ± 0.295	0.040 ± 0.168	-0.013 ± 0.192
pin-SVM	$w(1)$	2.276 ± 0.817	2.104 ± 0.839	2.198 ± 0.510	2.159 ± 0.524
$\tau = 0.2$	b	-0.012 ± 0.243	0.034 ± 0.280	0.019 ± 0.186	0.006 ± 0.208
pin-SVM	$w(1)$	2.188 ± 0.876	1.950 ± 0.947	2.114 ± 0.512	2.005 ± 0.499
$\tau = 0.1$	b	-0.035 ± 0.237	0.015 ± 0.301	0.032 ± 0.203	-0.112 ± 0.226
C-SVM	$w(1)$	1.498 ± 1.012	1.227 ± 0.960	1.665 ± 0.741	1.306 ± 0.583
	b	-0.013 ± 0.379	0.099 ± 0.485	-0.044 ± 0.222	0.012 ± 0.416
TSVC	$w(1)$	2.071 ± 0.662	1.584 ± 0.750	1.746 ± 0.642	1.606 ± 0.583
	b	-0.084 ± 0.404	-0.011 ± 0.410	-0.138 ± 0.143	-0.002 ± 0.323

In [2], one synthetic data set is provided, denoted as Data 2. In this data set, the classification boundary is $x(1) - x(2) = 0$ and the training data are taken from the uniform distribution on $[-5, 5]^2$ with the above noise. The average classification accuracy for 20 trials is reported in the third and the fourth line of Table III. For this data set, the C-SVM and the TSVC perform better than the pin-SVM. For Data 1, the features of one class come from a Gaussian distribution. In Data 2, the features follow the uniform distribution. For Gaussian distribution or any distribution with high probability density in its center, it is reasonable to pursue a small within-class scatter and hence the pinball loss minimization can tolerate noise and give a good result. But for the uniform distribution, small scatter does not lead to a good performance and the C-SVM works better for such problems.

B. UCI data sets with noise

The above experiments on synthetic data imply that the pin-SVM is less sensitive to noise around the decision boundary, especially when the features (or the mapped features in high dimension space) of each class have centralization property. In many applications, the features have this property, and hence we can expect the performance of the pin-SVM. In the following, real world data downloaded from UCI Repository of Machine Learning Dataset [34] are tested. In the first four data sets, i.e., "Monk1", "Monk2", "Monk3" and "Spect", training and testing sets are provided. For the others, we randomly partition the data into two parts, one of which is used for training (containing half of the data) and the other one is for testing. We let the features corrupted by zero-mean Gaussian noise. The training and the testing data are corrupted by the same noise. For each feature, the ratio of the variance of noise to that of the feature, denoted as r, is set to be 0 (i.e., noise-free), 0.05, and 0.1. Notice that except for the first four data sets, the training data are randomly selected and the experiments for these data sets also contain re-sampling factor.

We select the RBF kernel and apply the C-SVM, the pin-SVM, and the TSVC to do classification. To exclude the effect of the hyper-parameters C and σ, the same values are used for the three methods. We apply the least squares SVM (i.e., SVM with ℓ_2 loss and equality constraints, see [31]), which can be solved very efficiently, to tune the parameters based on 10-fold cross validation and grid search. The obtained values are then used in the C-SVM, the pin-SVM, and the TSVC. For each data set and each r, the experiments are repeated 10 times. The average and the standard deviation of the classification accuracy on the testing data are reported in Table IV, where the best ones in the view of accuracy are written in bold.

Generally, the pin-SVM achieves satisfactory results, that means the average accuracy is high and the standard deviation is small. A small deviation means that the pin-SVM is not sensitive to noise, which supports the theoretical analysis. One noticeable data is "Spect", for which the precision of the pin-

TABLE IV
CLASSIFICATION ACCURACY PERCENTAGE ON TESTING DATA FOR UCI DATASETS

Data Name	r	pin-SVM $\tau=1$	$\tau=0.5$	$\tau=0.2$	$\tau=0.1$	C-SVM	TSVC
Monk1	0.00	86.99 ± 0.00	**87.15** ± 0.00	87.08 ± 0.00	83.61 ± 0.00	82.52 ± 0.00	82.68 ± 0.00
	0.05	81.18 ± 1.97	84.15 ± 2.26	82.80 ± 2.37	**84.95** ± 2.28	77.75 ± 4.90	76.11 ± 2.21
	0.10	77.68 ± 1.61	77.39 ± 1.70	77.17 ± 1.52	**77.31** ± 2.58	74.31 ± 4.02	71.90 ± 1.93
Monk2	0.00	**87.41** ± 0.00	**87.41** ± 0.00	**87.41** ± 0.00	**87.41** ± 0.00	87.29 ± 0.00	87.29 ± 0.00
	0.05	82.27 ± 1.60	**85.27** ± 2.06	**85.27** ± 1.89	82.27 ± 1.60	81.92 ± 2.04	81.90 ± 1.09
	0.10	75.95 ± 2.10	**77.95** ± 2.10	**77.95** ± 2.01	75.95 ± 2.10	72.50 ± 2.54	74.68 ± 2.19
Monk3	0.00	**95.46** ± 0.00	95.37 ± 0.00	94.91 ± 0.00	93.15 ± 0.00	91.85 ± 0.00	92.31 ± 0.00
	0.05	**93.07** ± 1.50	93.01 ± 2.53	92.35 ± 2.47	89.58 ± 1.77	87.58 ± 4.14	85.42 ± 2.52
	0.10	90.01 ± 1.61	**90.60** ± 2.06	89.78 ± 2.81	86.67 ± 1.93	85.03 ± 3.74	84.13 ± 2.06
Spect	0.00	82.11 ± 0.00	82.35 ± 0.00	81.82 ± 0.00	82.62 ± 0.00	**86.63** ± 0.00	83.16 ± 0.00
	0.05	81.82 ± 3.69	81.66 ± 3.80	81.66 ± 3.84	81.82 ± 5.67	**82.03** ± 5.26	82.03 ± 2.57
	0.10	**81.12** ± 4.35	80.64 ± 5.18	80.86 ± 5.68	80.91 ± 5.61	80.69 ± 6.45	80.96 ± 4.65
Pima	0.00	75.99 ± 1.14	76.75 ± 1.08	76.81 ± 1.13	**76.83** ± 1.29	76.49 ± 1.45	74.74 ± 1.29
	0.05	76.47 ± 1.26	77.01 ± 1.14	77.32 ± 0.78	**77.58** ± 1.16	75.77 ± 1.98	77.58 ± 1.71
	0.10	**75.43** ± 1.17	75.24 ± 1.32	74.83 ± 2.05	75.25 ± 1.83	75.18 ± 3.32	74.31 ± 1.47
Breast	0.00	96.09 ± 0.77	96.11 ± 0.62	96.09 ± 0.74	96.09 ± 0.74	**96.20** ± 0.59	**96.20** ± 0.59
	0.05	96.03 ± 0.53	**96.11** ± 0.81	95.94 ± 0.61	95.71 ± 0.80	95.51 ± 0.86	95.60 ± 0.85
	0.10	**95.69** ± 0.72	95.77 ± 0.78	95.77 ± 0.80	95.83 ± 0.76	95.63 ± 1.37	95.66 ± 1.30
Trans.	0.00	77.84 ± 1.55	**78.98** ± 1.54	78.91 ± 1.47	78.22 ± 1.44	78.27 ± 1.59	75.18 ± 1.56
	0.05	77.12 ± 0.98	77.12 ± 1.14	77.20 ± 1.21	**77.39** ± 1.12	76.92 ± 1.94	75.74 ± 1.68
	0.10	**77.15** ± 1.42	**77.15** ± 1.52	77.07 ± 1.31	76.04 ± 1.37	76.90 ± 2.48	76.34 ± 2.22
Haberman	0.00	**72.99** ± 1.10	72.40 ± 1.66	72.21 ± 2.03	71.82 ± 1.96	72.14 ± 1.90	66.36 ± 1.44
	0.05	**73.12** ± 2.48	73.02 ± 2.19	72.91 ± 2.43	72.75 ± 2.40	71.75 ± 2.46	67.23 ± 1.66
	0.10	73.12 ± 2.90	73.12 ± 2.86	**73.18** ± 2.89	72.99 ± 2.92	71.25 ± 2.70	68.18 ± 1.84
Ionosphere	0.00	93.92 ± 1.04	**94.03** ± 1.20	93.92 ± 1.26	93.92 ± 1.36	93.69 ± 2.00	93.52 ± 1.21
	0.05	93.61 ± 2.09	93.70 ± 2.16	93.70 ± 2.17	**93.78** ± 2.11	92.58 ± 2.06	92.56 ± 1.96
	0.10	93.85 ± 1.70	93.80 ± 1.64	93.80 ± 1.63	**93.96** ± 1.76	91.09 ± 1.92	92.91 ± 1.91
Statlog	0.00	**84.04** ± 2.92	83.68 ± 2.10	83.46 ± 2.33	82.79 ± 2.71	82.01 ± 2.76	82.87 ± 2.94
	0.05	82.30 ± 2.75	82.60 ± 2.59	**82.93** ± 2.67	82.33 ± 2.56	81.26 ± 2.47	82.18 ± 2.47
	0.10	79.71 ± 3.33	**80.24** ± 2.48	79.95 ± 4.04	79.95 ± 3.88	79.58 ± 3.80	79.52 ± 3.78
Magic	0.00	82.29 ± 0.81	82.53 ± 0.92	83.02 ± 0.83	83.11 ± 0.80	**83.25** ± 1.42	83.16 ± 1.41
	0.05	80.19 ± 0.64	**80.96** ± 0.67	80.75 ± 0.52	80.79 ± 0.55	80.13 ± 0.87	78.82 ± 1.31
	0.10	78.77 ± 0.88	**79.04** ± 0.76	78.41 ± 2.82	78.61 ± 2.71	78.76 ± 2.51	76.75 ± 1.41
Spambase	0.00	89.52 ± 0.63	89.67 ± 0.65	89.84 ± 0.68	**89.93** ± 0.78	89.85 ± 0.97	89.83 ± 0.86
	0.05	88.90 ± 1.04	88.97 ± 1.07	89.04 ± 1.12	**89.09** ± 1.12	88.36 ± 1.29	88.37 ± 1.19
	0.10	87.74 ± 0.82	87.79 ± 0.92	87.81 ± 1.00	**87.86** ± 0.95	87.12 ± 0.96	87.11 ± 0.96

SVM is worse than that of the C-SVM. To investigate the reason, we consider the data without noise and calculate

$$\hat{f}(x) = \sum_{i=1}^{m} \hat{\lambda}_i y_i K(x_i, x) + \hat{b}$$

for each point, where $\hat{\lambda}_i, \hat{b}$ are obtained by the C-SVM (8) or the pin-SVM (7). Then the kernel function based method is used to estimate the p.d.f. of $y\hat{f}(x)$. For one sampling trial, the p.d.f. of $y\hat{f}(x)$ obtained by the C-SVM and the pin-SVM with $\tau = 0.5$ are shown in Fig.10, where the results for "Monk1" are shown by blue solid lines, "Spect" by red dashed lines, and "Statlog" by green dash-dotted lines.

From Fig.10, we observe that the results of the pin-SVM are grouped close to 1, which coincides with the analysis in Section III-D. Generally, small scatter leads to good classification performance. For example, the performance of the pin-SVM is better than that of the C-SVM for data set "Monk1" and "Statlog". However, for data set "Spect", the p.d.f. of $y\hat{f}(x)$ obtained by the C-SVM has two peaks, indicating that there exist multiple subclasses in class **I** or **II**. In this case, it is not reasonable to pursue a small scatter and the result of the pin-SVM is not good. We can tune τ for a better result: in this trial, when $\tau = 0.5$, the testing accuracy is 0.824, when $\tau = 0.05$, the accuracy is 0.845, and when $\tau = 0$ (i.e., C-SVM), the accuracy is 0.866. In regular binary classification problems, there do not exist subclasses and simply setting $\tau = 0.5$ or 0.2 gets good perform, as reported in Table IV.

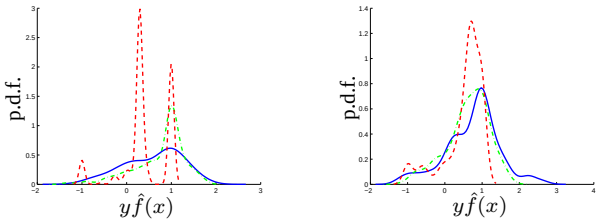


Fig. 10. P.d.f. of $yf(x)$ obtained by (a) the hinge loss SVM; (b) the pinball loss SVM with $\tau = 0.5$. The results for "Monk1" are shown by blue solid lines, "Spect" by red dashed lines, and "Statlog" by green dash-dotted lines.

TABLE V
CLASSIFICATION ACCURACY AND THE NUMBER OF NONZERO DUAL VARIABLES FOR THE PIN-SVM ($\tau = 0.5$)

ε	Monk1	Monk2	Monk3	Spect	Pima	Breast
0.0	84.15 (124)	85.27 (169)	93.01 (169)	82.35 (80)	77.01 (374)	96.11 (350)
0.05	81.25 (113)	84.40 (164)	92.85 (109)	82.52 (74)	74.49 (328)	96.54 (194)
0.1	80.21 (110)	84.86 (161)	92.94 (101)	83.65 (57)	75.30 (291)	95.77 (123)
0.2	78.77 (96)	82.78 (154)	91.20 (84)	83.44 (57)	76.26 (251)	95.54 (102)
C-SVM	77.75 (61)	81.92 (157)	87.58 (38)	86.63 (69)	75.77 (222)	95.51 (82)
ε	Trans.	Haberman	Ionosphere	Statlog	Magic	Spambase
0.0	77.89 (374)	73.02 (153)	93.70 (176)	82.60 (135)	80.96 (500)	88.97 (500)
0.05	76.64 (218)	72.79 (97)	93.52 (155)	82.06 (120)	80.53 (403)	87.48 (428)
0.1	76.93 (199)	72.86 (86)	93.60 (144)	81.16 (109)	79.59 (359)	88.08 (376)
0.2	77.32 (185)	72.40 (82)	93.01 (136)	81.10 (90)	79.53 (292)	88.16 (311)
C-SVM	76.92 (192)	71.75 (80)	92.58 (134)	81.26 (74)	80.13 (254)	88.36 (222)

C. Sparsity of the insensitive zone pinball loss SVM

In this part, we evaluate the effect of introducing insensitive zone to the pin-SVM. We set $\tau = 0.5$ and use the same σ and C obtained previously, then the ε insensitive zone pinball loss SVMs (18) with $\varepsilon = 0.05, 0.1,$ and 0.2 are applied to the considered data. The average classification accuracy of 20 trials and the number of nonzero dual variables are reported in Table V, where the results of the C-SVM are given for reference as well. Generally, with the increase of ε, the result becomes more sparse but the accuracy decreases. This observation helps us find a suitable value for ε. There is an exception for data set "Spect", for which the identification accuracy increases when ε becomes large. In the ε insensitive zone pin-SVM, there is no penalty on the points located in the insensitive zone, i.e., we do not care about the scatter of these points. When ε is large, the two peaks shown in Fig.10(a) are covered by the insensitive zone and have no effect on the result, which may lead to a high classification accuracy.

VI. CONCLUSION

Motivated by the link between the pinball loss and quantile regression, we introduced the pinball loss to classification problems, resulting in the pinball loss SVM classifier. The dual formulation of the pin-SVM is given and then positive definite kernels are applicable. The difference between L_τ and $L_{\rm hinge}$ is that the pinball loss gives penalty on the correctly classified points. We showed that such penalty does not change some crucial properties. For example, the function obtained by minimizing L_τ is the Bayes classifier and the classification error bound for L_τ preserves the same as that of $L_{\rm hinge}$. Meanwhile, the computational complexity of the pin-SVM is similar to that of the hinge loss SVM.

Compared to the hinge loss SVM, the major advantage of the proposed method is that the pin-SVM is less sensitive to noise, especially the feature noise around the decision boundary. This can be interpreted from the optimality condition or from the view of scatter minimization. According to a similar reason, the pin-SVM is more stable than the hinge loss SVM for re-sampling. Both the noise insensitivity and the re-sampling stability are supported by numerical experiments. Generally, the proposed method provides a promising tool for noise corrupted data. There are also some potential problems for further study, including sequential optimization methods and re-sampling based algorithms for large scale problems.

ACKNOWLEDGMENT

The authors are grateful to the anonymous reviewers for insightful comments.

REFERENCES

[1] V. Vapnik, *The Nature of Statistical Learning*. Springer-Verlag, 1995.
[2] J. Bi and T. Zhang, "Support vector classification with input data uncertainty," in *Neural Information Processing Systems (NIPS)*, vol. 17, 2004, pp. 161–168.
[3] I. Guyon, N. Matic, and V. Vapnik, "Discovering informative patterns and data cleaning," *Advances in Knowledge Discovery and Data Mining*, pp. 181–203, 1996.
[4] R. Herbrich and J. Weston, "Adaptive margin support vector machines for classification," in *the 9th International Conference on Artificial Neural Networks*, vol. 2, 1999, pp. 880–885.
[5] Q. Song, W. Hu, and W. Xie, "Robust support vector machine with bullet hole image classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 32, no. 4, pp. 440–448, 2002.
[6] W. Hu and Q. Song, "An accelerated decomposition algorithm for robust support vector machines," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 51, no. 5, pp. 234–240, 2004.
[7] G. Lanckriet, L. Ghaoui, C. Bhattacharyya, and M. Jordan, "A robust minimax approach to classification," *The Journal of Machine Learning Research*, vol. 3, pp. 555–582, 2003.
[8] P. Shivaswamy, C. Bhattacharyya, and A. Smola, "Second order cone programming approaches for handling missing and uncertain data," *The Journal of Machine Learning Research*, vol. 7, pp. 1283–1314, 2006.
[9] H. Xu, C. Caramanis, and S. Mannor, "Robustness and regularization of support vector machines," *The Journal of Machine Learning Research*, vol. 10, pp. 1485–1510, 2009.
[10] X. Zhang, "Using class-center vectors to build support vector machines," in *IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing*, 1999, pp. 3–11.
[11] J. Feng and P. Williams, "The generalization error of the symmetric and scaled support vector machines," *IEEE Transactions on Neural Networks*, vol. 12, no. 5, pp. 1255–1260, 2001.
[12] M. Yoon, Y. Yun, and H. Nakayama, "A role of total margin in support vector machines," in *the International Joint Conference on Neural Networks*, vol. 3, 2003, pp. 2049–2053.
[13] B. Jin, Y. Tang, and Y. Zhang, "Support vector machines with genetic fuzzy feature transformation for biomedical data classification," *Information Sciences*, vol. 177, no. 2, pp. 476–489, 2007.
[14] P. Lingras and C. Butz, "Rough set based 1-v-1 and 1-v-r approaches to support vector machine multi-classification," *Information Sciences*, vol. 177, no. 18, pp. 3782–3798, 2007.
[15] J. Zhang and Y. Wang, "A rough margin based support vector machine," *Information Sciences*, vol. 178, no. 9, pp. 2204–2214, 2008.
[16] R. Koenker, *Quantile Regression*. Cambridge University Press, 2005.

[17] I. Steinwart and A. Christmann, "How SVMs can estimate quantiles and the median," in *Neural Information Processing Systems (NIPS)*, vol. 20, 2007, pp. 305–312.
[18] ——, "Estimating conditional quantiles with the help of the pinball loss," *Bernoulli*, vol. 17, no. 1, pp. 211–225, 2011.
[19] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.
[20] A. Christmann and I. Steinwart, "On robustness properties of convex risk minimization methods for pattern recognition," *The Journal of Machine Learning Research*, vol. 5, pp. 1007–1034, 2004.
[21] F. Cucker and D. Zhou, *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, 2007.
[22] I. Steinwart and A. Christmann, *Support Vector Machines*. Springer-Verlag, 2008.
[23] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting algorithms as gradient descent in function space," in *Neural Information Processing Systems (NIPS)*, vol. 12, 1999, pp. 512–518.
[24] Y. Wu and Y. Liu, "Robust truncated hinge loss support vector machines," *Journal of the American Statistical Association*, vol. 102, no. 479, pp. 974–983, 2007.
[25] Y. Lin, "A note on margin-based loss functions in classification," *Statistics and Probability Letters*, vol. 68, no. 1, pp. 73–82, 2004.
[26] P. Bartlett, M. Jordan, and J. McAuliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006.
[27] R. Nock and F. Nielsen, "Bregman divergences and surrogates for learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2048–2059, 2009.
[28] M. Reid and R. Williamson, "Information, divergence and risk for binary experiments," *Journal of Machine Learning Research*, vol. 12, pp. 731–817, 2011.
[29] T. Zhang, "Statistical analysis of some multi-category large margin classification methods," *The Journal of Machine Learning Research*, vol. 5, pp. 1225–1251, 2004.
[30] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
[31] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.
[32] T. Van Gestel, J. A. K. Suykens, G. Lanckriet, A. Lambrechts, B. De Moor, and J. Vandewalle, "Bayesian framework for least-squares support vector machine classifiers, gaussian processes, and kernel fisher discriminant analysis," *Neural Computation*, vol. 14, no. 5, pp. 1115–1147, 2002.
[33] V. Jumutc, X. Huang, and J. A. K. Suykens, "Fixed-size pegasos for hinge and pinball loss SVM," in *the International Joint Conference on Neural Networks*, 2013, pp. 1122–1128.
[34] A. Frank and A. Asuncion, "UCI Machine Learning Repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

Xiaolin Huang received the B.S. degree in control science and engineering, and the B.S. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China in 2006. In 2012, he received the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China. Since then, he has been working as a postdoctoral researcher in ESAT-SCD-SISTA, KU Leuven, Leuven, Belgium. His current research areas include optimization, classification, and identification for nonlinear systems via piecewise linear analysis.

Lei Shi received the B.S. degree in information and computing science from Hefei University of Technology, Hefei, China in 2004. In 2010, he received the Ph.D. degrees in applied mathematics from University of Science and Technology of China, Hefei, China, and from City University of Hong Kong, Hong Kong, China. He is now a lecturer in School of Mathematical Sciences, Fudan University, Shanghai, China. From July 8th, 2012 to July 7th, 2013, he works as a postdoctoral researcher in ESAT-SCD-SISTA, KU Leuven, Leuven, Belgium. His research interests include statistical learning theory and approximation theory.

Johan A.K. Suykens was born in Willebroek Belgium, on May 18 1966. He received the M.S. degree in Electro-Mechanical Engineering and the Ph.D. Degree in Applied Sciences from the Katholieke Universiteit Leuven, in 1989 and 1995, respectively. In 1996 he has been a Visiting Postdoctoral Researcher at the University of California, Berkeley. He has been a Postdoctoral Researcher with the Fund for Scientific Research FWO Flanders and is currently a Professor (Hoogleraar) with KU Leuven.

Prof. J.A.K. Suykens is author of the books Artificial Neural Networks for Modelling and Control of Non-linear Systems (Kluwer Academic Publishers) and Least Squares Support Vector Machines (World Scientific), co-author of the book Cellular Neural Networks, Multi-Scroll Chaos and Synchronization (World Scientific) and editor of the books Nonlinear Modeling: Advanced Black-Box Techniques (Kluwer Academic Publishers) and Advances in Learning Theory: Methods, Models and Applications (IOS Press). In 1998 he organized an International Workshop on Nonlinear Modeling with Time-series Prediction Competition. He is a Senior IEEE member and has served as associate editor for the IEEE Transactions on Circuits and Systems (1997-1999 and 2004-2007) and for the IEEE Transactions on Neural Networks (1998-2009). He received an IEEE Signal Processing Society 1999 Best Paper (Senior) Award and several Best Paper Awards at International Conferences. He is a recipient of the International Neural Networks Society INNS 2000 Young Investigator Award for significant contributions in the field of neural networks. He has served as a Director and Organizer of the NATO Advanced Study Institute on Learning Theory and Practice (Leuven 2002), as a program co-chair for the International Joint Conference on Neural Networks 2004 and the International Symposium on Nonlinear Theory and its Applications 2005, as an organizer of the International Symposium on Synchronization in Complex Networks 2007 and a co-organizer of the NIPS 2010 workshop on Tensors, Kernels and Machine Learning. He has been awarded an ERC Advanced Grant 2011.